Explicit Proofs and The Flip

Dedicated to Sri Ramakrishna

Ketan D. Mulmuley *
The University of Chicago

September 14, 2010

Abstract

This article describes a formal strategy of geometric complexity theory (GCT) to resolve the *self referential paradox* in the P vs. NP and related problems. The strategy, called the *flip*, is to go for *explicit proofs* of these problems. By an explicit proof we mean a proof that constructs proof certificates of hardness that are easy to verify, construct and decode. The main result in this paper says that (1) any proof of the arithmetic implication of the P vs. NP conjecture is close to an explicit proof in the sense that it can be transformed into an explicit proof by proving in addition that arithmetic circuit identity testing can be derandomized in a blackbox fashion, and (2) stronger forms of these arithmetic hardness and derandomization conjectures together imply a polynomial time algorithm for a formidable explicit construction problem in algebraic geometry. This may explain why these conjectures, which look so elementary at the surface, have turned out to be so hard.

1 Introduction

Geometric complexity theory (GCT) is an approach to fundamental hardness problems in complexity theory via algebraic geometry and representation theory suggested in a series of articles [22]-[29], which we call GCT1-8. In this article we describe and justify a formal defining strategy of GCT, called the *flip*, to resolve the *self referential paradox* in the P vs. NP and related problems. This paradox refers to the question that is often asked: namely, since the P vs. NP problem is a universal statement about mathematics that says that discovery is hard, why could it not preclude its own proof and hence be independent of the axioms of set theory? Resolution of this self referential paradox is generally regarded as the root difficulty in this problem; cf. the survey [2] and the references therein.

The flip strategy of GCT to resolve the self referential paradox is to go for an *explicit proof*. By an *explicit proof* of the nonuniform P vs. NP problem (i.e., $NP \not\subseteq P/poly$ conjecture) we essentially mean a proof that shows existence of proof certificates for hardness of an NP-complete function $f(X) = f(x_1, \ldots, x_n)$, also called *obstructions* (to efficient computation of $f(X)$), that

*Supported by NSF grant CCF-1017760.

are *short* (of poly(n) bitsize), *easy to verify and construct* (in poly(n) time), and *easy to decode*. By easy to decode we mean that, given n, small m = poly(n), and a short obstruction s, a small set $S_{n,m}(s) = \{X_1, \ldots, X_r\}$, r = poly(n), of inputs can be constructed in poly(n) time such that, for every small circuit C of size $\leq m$, $S_{n,m}(s)$ contains a counterexample X_C such that $f(X_C) \neq C(X_C)$. Here $C(X)$ denotes the function computed by C. A proof technique that yields an explicit proof of the nonuniform P vs. NP problem is called a *flip* (from hard to easy), because in essence it reduces the original hardness (lower bound) problem to easiness (upper bound) problems: namely, to showing that verification, construction and decoding of proof certificates of hardness as per that technique are easy, i.e., belong to the complexity class P. In what sense this strategy amounts to an explicit resolution of the self referential paradox is explained in Section 3.3. See Section 3 for the definition of the flip in the arithmetic setting.

The main results in this article provide a posteriori [1] justification for this flip strategy. Specifically, it is shown (cf. Flip Theorems 4.2 and 4.3) that any proof of the arithmetic nonuniform version of the P vs. NP conjecture in GCT1 (which is a formal weaker implication of the boolean $NP \not\subseteq P/poly$ conjecture) can be converted into an explicit proof by proving in addition that circuit identity testing can be derandomized in a blackbox fashion. This standard derandomization assumption [32, 12, 13] is generally believed to be easier than the target lower bound. Hence, in this sense, any proof of the arithmetic P vs. NP conjecture is close to an explicit proof. It is also shown (cf. Flip Theorem 9.2) that stronger forms of these arithmetic hardness and derandomization conjectures together imply a polynomial time algorithm for a formidable explicit construction problem in algebraic geometry. This may explain why these conjectures in complexity theory, which look so elementary at the surface, have turned out to be so hard.

A starting point for the investigation in this article was an analogous result (cf. Flip Lemma 4.1) for (weak) arithmetic hardness of the permanent that follows easily from the hardness vs. randomness principle [12, 13] and downward self reducibility of the permanent. Specifically, it follows by derandomizing the co-RP algorithm in [12] for testing if a given arithmetic circuit computes the permanent using its downward self reducibility. But self-reducibility does not seem to be as effective in the context of the P vs. NP problem, as has already been observed in other contexts in complexity theory (e.g. average vs. worst case hardness [4, 6]). The best earlier results in the context of the P vs. NP problem were proved in [3, 7]. Using downward self reducibility, the article [3] gives, assuming $NP \not\subseteq P/poly$, a probabilistic polynomial time algorithm for finding, given any small circuit C, a counterexample on which it differs from SAT. But this algorithm cannot efficiently produce a small set (a proof certificate of hardness) that contains a counterexample against *every* small circuit. The article [7] gives under the same assumption a probabilistic polynomial time algorithm with an access to the SAT oracle for computing a small set of satisfiable formulae that contains a counterexample against *every* small circuit claiming to compute SAT. The main difficulty in the context of the P vs. NP problem is to accomplish the same task in polynomial time under reasonable complexity

[1] This strategy was formulated in a rough form after the completion of GCT1 and 2, when it was realized that these initial papers do not address the self referential paradox. It was announced briefly without any explanations in [20]. The articles GCT3-5 investigate some basic problems in representation theory motivated by the flip, and the main result of GCT6, based on GCT1-5 and other results in algebraic geometry and representation theory, provides an approach to implement the flip in the arithmetic setting wherein the underlying field of computation has characteristic zero.

theoretic assumptions without any access to the SAT oracle. This difficulty is overcome here in the setting of the arithmetic P vs. NP conjecture using the hardness vs. randomness principle [32, 12, 13] in conjunction with characterization by symmetries of a certain exceptional function associated with the complexity class NP in GCT1 (cf. Section 5). Characterization by symmetries is a well known phenomenon in invariant theory on which GCT is based. Its crucial role here suggests that it may find more applications in complexity theory in future.

The flip lemma (Lemma 4.1) for the weak arithmetic hardness of the permanent also does not have any direct implications in algebraic geometry, unlike the stronger flip theorem (Theorem 9.2). This stronger theorem is proved by combining the hardness vs. randomness principle and characterization by symmetries with classical algebraic geometry.

There is also a flip theorem in the boolean setting (Flip Theorem 10.5) for a stronger average case form of the boolean $NP \not\subseteq P/poly$ conjecture based on the characterization by symmetries. The main ingradient here is just the formulation of this conjecture. The rest follows easily from the work [32, 12] on derandomization of BPP. All the nonuniform results in this article also have analogues in the uniform setting.

In view of all these results, the flip strategy of GCT to go for explicit proofs of the P vs. NP and related conjectures seems quite natural.

The rest of this article is organized as follows. Section 2 describes the arithmetic version of the P vs. NP problem defined in GCT1. Section 3 describes the formal flip strategy for resolution of the self referential paradox via explicit proofs. The flip theorems in various arithmetic settings are stated in Section 4 and proved in Sections 5-8. The implication in algebraic geometry is pointed out in Section 9. The flip theorem in the boolean setting is stated in Section 10.

No familiarity with algebraic geometry is assumed in this paper. The required facts from classical algebraic geometry are only used as blackboxes.

2 Arithmetic versions of the P vs. NP and related problems

In this section we recall the arithmetic version of the P vs. NP problem defined in GCT1 and also arithmetic versions of the related problems.

2.1 Arithmetic hardness of the permanent

By the arithmetic hardness conjecture for the permanent, we mean the problem of showing that the permanent of an $n \times n$ complex matrix X cannot be computed by any arithmetic circuit over $\mathbb{C}$ of $m = \text{poly}(n)$ size, where by the size of the circuit we mean the total number of nodes in it. By the weak arithmetic hardness conjecture, we mean the problem of showing that the permanent of an $n \times n$ integer matrix X cannot be computed by any arithmetic circuit (over Z or $\mathbb{Q}$) of $m = \text{poly}(n)$ total bit size, where by the total bit size of the circuit we mean the total number of nodes in it plus the the total bit size of all constants in the circuit. Clearly, the weak arithmetic conjecture is implied by the arithmetic conjecture. By the strong arithmetic conjecture [cf. GCT1], we mean the problem of showing that $\text{perm}(X)$, the permanent of an $n \times n$ variable matrix X, cannot be approximated infinitesimally closely by an arithmetic circuit over $\mathbb{C}$ of $m = \text{poly}(n)$ size. Here by infinitesimally close approximation, we mean that all coefficients of the polynomial computed by a circuit can be made infinitesimally close to that of the permanent. Clearly, the strong arithmetic conjecture implies the arithmetic conjecture.

By the arithmetic permanent vs. determinant problem [36], we mean the problem of showing that perm(X) cannot be represented linearly as det(Y), the determinant of an $m \times m$ matrix Y, if $m = \text{poly}(n)$, or more generally, $m = 2^{\log^a n}$, for a fixed constant $a > 0$, and $n \to \infty$; the best known lower bound on m at present is quadratic [19]. Here, by a linear representation, we mean that the entries of Y are (possibly nonhomogeneous) linear functions (over $\mathbb{C}$) of the entries of X. The strong arithmetic version of this problem [GCT1] is to show that perm(X) cannot be approximately infinitesimally closely by an expression of the form det(Y) as above. Clearly, the strong arithmetic version implies that arithmetic version. The current best lower bound in the strong arithmetic setting is quadratic. It is proved in [14] using GCT, and provides the first concrete lower bound application of GCT in the context of the permanent vs. determinant problem. The weak arithmetic version of this problem is to show that perm(X) cannot be represented linearly as det(Y), where the entries of Y are possibly nonhomogeneous linear functions over $\mathbb{Z}$ and the total bit size of the specification of Y is poly(n), or more generally, $O(2^{\log^a n})$, for a fixed constant $a > 0$. Clearly the weak arithmetic version is implied by the arithmetic version.

A priori, it is not at all clear that the strong arithmetic conjectures above are actually stronger than the arithmetic conjectures. This is expected because there are functions that can be approximated infinitesimally closely by small circuits (of small depth) but conjecturally cannot be computed by small circuits (of small depth); cf. Section 4.2 in GCT1.

2.2 Arithmetic P vs. NP problem

Next we turn to the arithmetic version of the P vs. NP problem defined in GCT1. Towards that end, we first associate with the complexity class NP a certain integral function $E(X)$ that is characterized by its symmetries (cf. Section 5) like the permanent function associated with the complexity class $\#P$.

Take a set $\{X_i^j | 1 \leq j \leq k,\ 1 \leq i \leq m\}$ of m-dimensional vector variables, for some fixed constant $k \geq 3$. Here each X_i^j is an m-vector. So there are km vector variables overall. Let X be the $m \times km$ variable matrix whose columns consist of these km variable vectors. For any function $\sigma : \{1, \ldots, m\} \to \{1, \ldots, k\}$, let $\det_\sigma(X)$ denote the determinant of the matrix X_σ whose i-th column is $X_i^{\sigma(i)}$. Define $E(X) = \prod_\sigma \det_\sigma(X)$ where σ ranges over all such functions. Clearly $E(X)$ is well defined over any base field F. Let $n = km^2$ be the total number entries in X.

The ultimate goal of GCT is:

Conjecture 2.1 (The stronger form of the $NP \not\subseteq P/poly$ conjecture) *Let the base field $F = F_p$, $p = q^l$, $q = O(poly(n))$ a prime and $l = n^a$, for a fixed constant $a > 1$. Then $E(X)^r$, for any $0 < r < p$, cannot be computed by an arithmetic circuit over F_p of poly(n) size.*

Here the rank l of F_p is required to be large so that the size of F_p is much larger than the degree mk^m of $E(X)$. Computation of $E(X)$ has been conjectured to be hard in this case because to decide whether $E(X)$ is zero over $\mathbb{Z}$ is known to be NP-complete (cf. page 451 in [10]).

Proposition 2.2 *(cf. Section 6) Conjecture 2.1 implies $NP \not\subseteq P/poly$.*

An intermediate goal is:

Conjecture 2.3 *[cf. GCT1]*

(a) [The (nonuniform) arithmetic P vs NP problem] Suppose the base field (or ring) $F = \mathbb{Q}$ or $\mathbb{C}$ (or $\mathbb{Z}$). Then $E(X)$ cannot be computed by an arithmetic circuit of poly(n) size over F.

(b) [The weak (nonuniform) arithmetic P vs. NP problem] $E(X)$ (over $\mathbb{Z}$) cannot be computed by an arithmetic circuit over $\mathbb{Z}$ of total bit size O(poly(n)).

(c) [The strong (nonuniform) arithmetic P vs. NP problem] $E(X)$ cannot be approximated infinitesimally closely by an arithmetic circuit (over $\mathbb{C}$) of size poly(n).

Here (b) over $\mathbb{Z}$ is a weaker implication of Conjecture 2.1 for $E(X)$ over F_p. It is also implied by the usual $NP \not\subseteq P/poly$ conjecture since, as already remarked, the problem of deciding if $E(X) = 0$ over $\mathbb{Q}$ or $\mathbb{Z}$ is NP-complete [10]. Furthermore, (b) is a weaker implication of (a), because in (a) there is no restriction on the bitlengths of the integer constants in the circuit computing $E(X)$. Only the total number of nodes in the circuit needs to be $O(\text{poly}(n))$. Whereas in (b) the total number of nodes as well as the total bit size of the constants in the circuit need to be $O(\text{poly}(n))$.

3 The flip and explicit proofs

In this section we describe the formal flip strategy towards the uniform or nonuniform P vs. NP and related problems in the boolean as well as arithmetic settings.

First let us consider the nonuniform boolean setting. Fix an NP-complete function $f(X) = f(x_1, \ldots, x_n)$, say SAT. The goal of the nonuniform P vs. NP problem (i.e., $NP \not\subseteq P/poly$ conjecture) is to show that there does not exist a small circuit C of size $m = \text{poly}(n)$ that computes $f(X)$, $n \to \infty$. Equivalently, the goal is to prove:

(HOH: Hard Obstruction Hypothesis): For every large enough n, and $m = \text{poly}(n)$, there exists a *trivial obstruction (i.e. a "proof-certificate" of hardness)* to the efficient computation of $f(X)$. Here by a trivial obstruction we mean a table that lists for every small circuit C a counterexample X such that $f(X) \neq C(X)$, where $C(X)$ denotes the function computed by $C(X)$.

The number of rows of this table is equal to the number of circuits of size $m = \text{poly}(n)$. Thus the size of this table is exponential; i.e., $2^{O(\text{poly}(n))}$. The time to verify whether a given table is a trivial obstruction is also exponential, and so is the time of the obvious algorithm to decide if such a table exists for given n and m, and to construct one if it exists. From the complexity theoretic viewpoint, this is a hard (inefficient) task. So we call this trivial, brute force strategy for proving the nonuniform P vs. NP conjecture, based on existence of trivial obstructions, a *hard strategy*–it is really just a restatement of the original problem. Hence, the terminology Hard Obstruction Hypothesis.

Any proof strategy for the P vs. NP problem has to answer the following question:

Question 3.1 *In what sense is the proof strategy fundamentally different from the trivial, brute force strategy above and not just an equivalent reformulation of the original problem? That is, in what sense are the proof certificates of hardness (obstructions) of this proof strategy fundamentally better than the trivial obstructions above?*

Until this question is answered, however sophisticated a proof strategy may be, it cannot be considered to be more than a restatement or an equivalent reformulation of the original problem.

The most obvious and natural abstract strategy that is fundamentally better than the trivial strategy is suggested by the P vs. NP problem itself. Before we define it, let us first see what is wrong with the trivial obstruction from the complexity-theoretic point of view. That is quite clear. First, it is long, i.e., its description takes exponential space. Second, it is hard to verify (and also construct); i.e., it takes exponential time. Since NP is the class of problems with "proof-certificates" that are short (of polynomial-size) and easy to verify (in polynomial-time), this then leads to the following strategy for proving the nonuniform $P \neq NP$ conjecture, based on proof certificates (obstructions) that are short, and easy to verify (and also easy to construct). We call this strategy *the flip*: from the hard (exponential time verifiable trivial obstructions) to the "easy" (polynomial time verifiable/constructible new obstructions), and from the nonexistence (lower bound problem) to the existence (upper bound problem)—existence of an efficient algorithm to verify and construct an obstruction.

Formally, we say that a technique for proving the nonuniform $P \neq NP$ conjecture (using the function $f(X)$) is a *flip* if there exists a family $\mathcal{O} = \cup_{m,n}\mathcal{O}_{n,m}$ of bit strings called *obstructions* (or obstruction labels), which serve as proof certificates of hardness of $f(X)$, having the following Flip properties F0-F4.

F0 [Short]: The set $\mathcal{O}_{n,m}$ is nonempty and contains a short obstruction string s if m is small, i.e., $m = O(\text{poly}(n))$, or more generally $m = O(2^{\log^a n})$, $a > 1$ a fixed constant. Here short means the bitlength $\langle s \rangle$ of s is poly(n, m). This is poly(n) if $m = \text{poly}(n)$.

To state F1, we define a *small global obstruction set* $S_{n,m}$ to efficient computation of $f(X)$, for given n and m, to be a small set $\{X_1, \ldots, X_l\}$, $l = \text{poly}(n, m)$, of inputs such that, for any circuit C of size $\leq m$, $S_{n,m}$ contains a counterexample $X_C = X_j$, for some $j \leq l$, such that $f(X_C) \neq C(X_C)$. Then:

F1 [Easy to decode]: Each bit string $s \in \mathcal{O}_{n,m}$, m small and s short, denotes a small global obstruction set $S_{n,m}(s)$ to efficient computation of $f(X)$ such that: (a) given s, n and m, $S_{n,m}(s)$ can be computed in poly$(\langle s \rangle, n, m)$ time–in particular, if s is short, $S_{n,m}(s)$ can be computed in poly(n, m) time–, and (b) given s, n, m and any circuit C of size $\leq m$, a set $S_{n,m,C}(s) \subseteq S_{n,m}(s)$ of $O(1)$ size can be computed in poly$(\langle s \rangle, n, m)$ time such that $S_{n,m,C}(s)$ contains some counterexample X_C such that $f(X_C) \neq C(X_C)$. A stronger form of (b) is (b'): given s, n, m and C, a counterexample $X_C \in S_{n,m}(s)$ as above can be computed in poly$(\langle s \rangle, n, m)$ time (we do not consider it in this paper).

F2 [Rich]: For every n and $m = \text{poly}(n)$, $\mathcal{O}_{n,m}$ contains at least $2^{\Omega(m)}$ pairwise disjoint obstructions, each of poly(n, m) bitlength. Here we say that two obstructions $s, s' \in \mathcal{O}_{n,m}$ are disjoint if $S_{n,m}(s)$ and $S_{n,m}(s')$ are disjoint.

F3 [Easy to verify]: Given n, m and a string s, whether s is a valid obstruction string for n and m–i.e., whether $s \in \mathcal{O}_{n,m}$–can be verified in poly$(n, \langle s \rangle, m)$ time. In particular, this time is poly(n) when $\langle s \rangle$ and m are $poly(n)$.

F4 [Easy to construct]: For each n and $m = \text{poly}(n)$ a valid obstruction string $s_{n,m} \in \mathcal{O}_{n,m}$ can be constructed in poly$(n, m) = \text{poly}(n)$ time.

This finishes the description of F0-4 defining a flip.

We say that a proof of the $NP \not\subseteq P/poly$ conjecture (using $f(X)$) is *extremely explicit* if it proves existence of an obstruction family $\mathcal{O}$ satisfying F0-4. We have defined explicitness in the most extreme form here, because we wish to prove the flip results later (Theorems 4.3 and 9.2) in a strongest possible form to indicate what is eventually possible. One may also consider weaker forms of explicitness (as we do in GCT) by relaxing the conditions above appropriately. We do not define them here since they are not used in this paper. Hence, in this paper, whenever we say explicit, we mean extremely explicit.

3.1 Uniform setting

Now let us consider the uniform setting. We say that a technique for proving the uniform $P \neq NP$ conjecture (using the function $f(X)$) is a (uniform) *flip*, and the resulting proof *explicit*, if there exists a family $\mathcal{O} = \cup_{m,n} \mathcal{O}_{n,m}$ of bit strings called *obstructions* (or obstruction labels), which serve as proof certificates of hardness of $f(X)$, satisfying the Uniform Flip properties UF0-UF4, which are obtained from F0-F4 by simply replacing the circuits in their definitions by uniform circuits. Note that UF1 (b) and UF4 together imply "efficient diagonalization within $O(1)$ factor": given $n, m = \text{poly}(n)$ and any algorithm C that works within m time on inputs of size n, a set $S_{n,m,C}$ of $O(1)$ size can be computed in poly(n, m) time such that $S_{n,m,C}$ contains some counterexample X_C such that $f(X_C) \neq C(X_C)$.

3.2 Arithmetic setting

We can similarly define the flip and explicit proofs for the arithmetic P vs. NP problem (Conjecture 2.3) letting $E(X)$ in Section 2 play the role of $f(X)$.

In the weak arithmetic setting, we replace boolean circuits of bit size $\leq m$ by arithmetic circuits of total bit size $\leq m$ in all definitions.

In the arithmetic setting, we replace boolean circuits of bit size $\leq m$ by arithmetic circuits of size (not bit size) $\leq m$ in all definitions. The obstructions in $\mathcal{O}_{n,m}$ are now meant to be against all arithmetic circuits of size $\leq m$. The running time bounds in all the definitions are the same as before except that the running time of the decoding algorithm in F1 (b) is meant to be poly$(n, m, \langle s \rangle)$, assuming unit-cost access to the circuit C as an oracle; the actual cost of evaluating C can be much larger than m now since there is no bound on the sizes of the constants in C. In the arithmetic setting we will mainly be interested in explicit proofs that have the following additional geometric property G.

To define it, we need some notation. For given $s \in \mathcal{O}_{n,m}$, let $S_{n,m}(s) = \{X_1, \ldots, X_l\}$,

$l = \text{poly}(n, m)$, denote the small glbal obstruction set as in F1 (a). Let V denote the space of polynomial functions in X of degree $\leq 2^m$. Thus the polynomial function $C(X)$ computed by any arithmetic circuit C of size $\leq m$ belongs to V. Let $\Sigma = \Sigma_{n,m} = \{C(X)\} \subseteq V$, where C ranges over all such circuits. The function $E(X)$ also belongs to V assuming that $2^m > \deg(E(X))$. Let $\psi_s : V \to \mathbb{C}^l$ be the linear map such that, for any $g(X) \in V$ and any $i \leq l$,

$$\psi_s(g(X))_i = g(X_i).$$

In other words, $\psi_s(g(X))$ is simply the l-tuple of evaluations of $g(X)$ at various X_i's, and $\psi_s(g(X))_i$ denotes the i-th entry in this tuple. Clearly $\psi_s(E(X)) \notin \psi_s(\Sigma)$ by the definition of an obstruction. We call ψ_s an *explicit linear separator* associated with s. The geometric property G mentioned above is as follows.

G: The point $\psi_s(E(X))$ does not belong to the closure of $\psi_s(\Sigma_{n,m})$ (in the usual complex topology) for any $s \in \mathcal{O}_{n,m}$.

The motivation here is as follows. In GCT we are interested in showing existence of an obstruction using algebro-geometric techniques. If $\psi_s(E(X))$ belongs to the closure of $\psi_s(\Sigma)$ then any polynomial function that vanishes on $\psi_s(\Sigma)$ will also vanish on $\psi(E(X))$. Hence no algebro-geometric technique will be able to distinguish $\psi_s(E(X))$ from $\psi_s(\Sigma)$. The property G is meant to rule out such pathological geometric behaviour and ensure that the separator ψ_s is good geometrically.

The flip in the strong arithmetic setting is defined by making the following change in the definitions of F0-4 and G in the arithmetic seting: replace a circuit of size $\leq m$ (or rather the function computed by it) everywhere by a function that can be approximated infinitesimally closely by circuits of size $\leq m$.

We can similarly define the flip and an explicit proof for the various arithmetic versions of the permanent vs. determinant problem, replacing a circuit by a linear (determinantal) representation. We can also define these notions for other lower bound problems in complexity theory such as the P vs. NC problem.

3.3 Self-referential paradox

We now explain in what sense implementation of the flip amounts to explicit resolution of the self referential paradox, and why this is such a formidable challenge.

Towards this end, let us examine the properties F above more closely. For an obstruction $s \in \mathcal{O}_{n,m}$, let $S_{n,m}(s)$ denote the corresponding global obstruction set in F1 (a) that can be computed in polynomial time. To simplify the argument, let us replace F1 (b) by (b)'. The decoding algorithm in (b)' gives in polynomial time a counterexample $X_C \in S_{n,m}(s)$ for every small circuit C of size $\leq m$. Let $\tilde{S}_{n,m}(s)$ denote the trivial obstruction of exponential size that lists for every small C this X_C. Then $S_{n,m}(s)$ can be thought of as a polynomial size encoding (i.e., information theoretic compression) of the trivial obstruction $\tilde{S}_{n,m}(s)$.To verify a given row of $\tilde{S}_{n,m}(s)$, we have to check if $f(X_C) \neq C(X_C)$ for the C corresponding to that row. For general X_C, this cannot be done in polynomial time, assuming $P \neq NP$, since f is NP-complete. And yet F3 says that whether s is a valid obstruction, i.e., whether each of the *exponentially many* rows of $\tilde{S}_{n,m}(s)$ specifies a counterexample, can be verified in polynomial time. At the surface, this may seem impossible. It may seem as if to prove $P \neq NP$, we are trying to prove $P = NP$. This is why implementation of the flip is such a formidable challenge.

4 Main results

That leads one to ask: why should we then go for explicit proofs for the nonuniform $P \neq NP$ and related conjectures when just proving existence of some obstructions even nonconstructively suffices in principle? The reason is provided by the following results (Theorems 4.2 and 4.3) which say that any proof of the arithmetic nonuniform P vs. NP conjecture (Conjecture 2.3) can converted into an explicit proof by proving in addition that arithmetic circuit identity testing can be derandomized in a blackbox fashion. This standard derandomization assumption [12, 13] is generally regarded as easier than the target lower bound. Hence, in this sense, any proof of the arithmetic P vs. NP conjecture is close to an explicit proof.

4.1 Weak arithmetic setting

We begin with a preliminary lemma in the context of the weak arithmetic hardness of the permanent as a motivation.

Lemma 4.1 (Flip, nonuniform weak arithmetic) *Assume the weak arithmetic hardness conjecture for the permanent: specifically, that the permanent of an $n \times n$ integer matrix X cannot be computed by any arithmetic circuit (over $\mathbb{Q}$) of $m = poly(n)$ total bit size. Suppose also that the complexity class E (consisting of the problems that can be solved in exponential time) does not have subexponential size circuits (or less stringently, that black box polynomial identity testing [1, 13] can be derandomized; cf. Section 7.4). Then:*

(1) For every n and $m = poly(n)$, it is possible to compute in $poly(n,m) = poly(n)$ time a small set $S_{n,m} = \{X_1, \ldots, X_l\}$, $l = poly(n,m) = poly(n)$, of $n \times n$ integer matrices such that for every arithmetic circuit C of total bit size $\leq m$, $S_{n,m}$ contains a matrix X_C which is a counter example against C, i.e, such that $perm(X_C)$ is not equal to the value $C(X_C)$ computed by the circuit. The set $S_{n,m}$ is thus a small global obstruction set of $poly(n,m) = poly(n)$ size against all small circuits of total bit size $\leq m$.

(2): Furthermore, assuming a slight strengthening of the assumption that E does not have subexponential size circuits (Conjecture 7.2 given later), or less stringently, that black box polynomial identity testing can be derandomized (Section 7.4), weak arithmetic hardness of the permanent has an explicit proof. Specifically, there exists, for every n and $m = poly(n)$, a set $\tilde{\mathcal{O}}_{n,m}$ of obstructions (bit strings) satisfying F0-F4.

(3) Similar result holds for the weak arithmetic form of the permanent vs. determinant problem [36] over $\mathbb{Q}$, replacing the second assumption in (1) and (2) by its weaker version–derandomization of symbolic determinant identity testing [13].

Lemma 4.1 follows (cf. Section 7.1) from the hardness vs. randomness principle [12, 13] in conjunction with characterization of the permanent by its symmetries (cf. Section 5). A slightly weaker form of Lemma 4.1 (everything therein except F1 (b)) follows easily (cf. Section 7.1) by derandomizing [32, 12] the co-RP algorithm in [13] for testing if a given arithmetic circuit C computes the permanent using its downward self-reducibility. But we cannot prove an analogous result in the context of the P vs. NP problem using self reducibility alone. Using downward self reducibility, the article [3] gives, assuming $NP \not\subseteq P/poly$, a probabilistic polynomial time algorithm for finding, given any small circuit C, a counterexample on which it differs from SAT; but this algorithm cannot efficiently produce a small *global* obstruction set against *all* small circuits. The related article [7] shows under the same assumption that there exists a small global obstruction set of satisfiable formulae which contains, for every small circuit C, a counter example on which it differs from SAT. But the algorithm in [7] for finding this set works in probabilistic polynomial time assuming access to the SAT oracle. Getting rid of this access to the SAT oracle is the main problem in the context of the $NP \not\subseteq P/poly$ conjecture. It is solved in the weak arithmetic setting in the following result.

Theorem 4.2 (Flip, nonuniform weak arithmetic) *Result analogous to the one in Lemma 4.1 also holds for the weak arithmetic nonuniform P vs. NP problem (cf. Conjecture 2.3 (a)) with the integral function $E(X)$ defined in Section 2 playing the role of the permanent in Lemma 4.1.*

This is proved (cf. Section 7) by combining the hardess vs. randomness principle [32, 12] with the fact [GCT1] that the function $E(X)$ is also *characterized by its symmetries* just like the permanent (cf. Section 5).

4.2 Arithmetic setting

We now turn to the arithmetic setting.

Theorem 4.3 (Flip, nonuniform arithmetic) *(a) Assume the strong arithmetic hardness conjecture for the permanent, and the associated strong derandomization hypothesis (defined in Section 8.1). Then the strong arithmetic hardness conjecture for the permanent has an explicit proof having the properties F0-4 and G. If we only assume arithmetic hardness conjecture for the permanent, and the associated derandomization hypothesis (defined in Section 8.1), then the arithmetic hardness conjecture for the permanent has an explicit proof having the properties F0-4 (but G cannot be guaranteed).*

(b) Similar results holds for the strong arithmetic P vs. NP and permanent vs. determinant problems (cf. Section 2).

This is proved in Section 8 using the the hardness vs. randomness principle and the characterization by symmetries (to prove the properties F0-4) in conjunction with some classical algebraic geometry (to prove the property G).

Unlike Lemma 4.1 and Theorem 4.2, Theorem 4.3 has a direct implication in algebraic geometry. Specifically, it implies (cf. Theorem 9.2) that solutions to the strong arithmetic hardness and derandomization conjectures under consideration will lead to polynomial time algorithms for really formidable explicit construction problems in algebraic geometry.

The obstruction family $\tilde{\mathcal{O}}$ in Lemma 4.1, or Theorem 4.2 or 4.3 does not depend on the proof technique at all. This obstruction family is of no use in actually proving hardness of the permanent or $E(X)$ since the proof of its existence assumes this hardness. The challenge in the implementation of the flip is to prove existence of an alternative family $\mathcal{O}$ of obstructions having the flip properties *without resorting to any hardness assumptions*. The main result of GCT, proved in GCT6, extending the investigation in GCT1-5, gives an approach to implement the flip for the arithmetic form of the P vs. NP problem (Conjecture 2.3) and the permanent vs. determinant problem.

A flip theorem like the one above is meaningful only if the hardness conjecture under consideration is harder than the additional derandomization conjecture assumed in its statement. Otherwise, it will really be talking about the difficulty of this additional derandomization conjecture. Thus the flip Theorem 4.3 does not say anything in the context of the quadratic lower bound [19] in the permanent vs. determinant problem. Indeed, the known proof in [19] for this quadratic lower bound is far from explicit. Here the (analogous) flip theorem will talk about the difficulty of the derandomization conjecture.

4.3 Boolean setting

Analogue of Theorem 4.2 also holds in the boolean setting for a stronger average case form of the usual (boolean) $NP \not\subseteq P/poly$ conjecture based on the characterization by symmetries; cf. Section 10. The main new ingradient here is just formulation of this conjecture. The rest follows easily from the work [12] on derandomization of BPP.

4.4 Uniform setting

The following results follow by uniformizing the proofs of Lemma 4.1 and Theorem 4.2.

Lemma 4.4 (Flip, uniform) *Assume that the permanent of an $n \times n$ integer matrix cannot be computed by a uniform circuit of $m = poly(n)$ bit size and that black box polynomial identity testing can be derandomized (Section 7.4)–this is a uniform assumption. Then the uniform hardness conjecture under consideration has an explicit proof satisfying UF0-4; this, in particular, implies efficient diagonalization within $O(1)$ factor.*

Theorem 4.5 (Flip, uniform) *Similar result holds for the weak uniform arithmetic hardness of $E(X)$.*

Analogous results also hold in the arithmetic and strong arithmetic settings with appropriate definition of uniformity.

5 Characterization by symmetries

We now describe the phenomenon of characterization by symmetries on which the proof of the flip lemma and theorems are based.

5.1 Permanent vs. determinant problem

In the context of the permanent vs. determinant problem, this phenomenon is that the permanent and determinant, the functions that are complete and almost complete for the complexity classes $\#P$ and NC, respectively, are *exceptional*, by which we mean they are characterized by their symmetries in the following sense.

Let Y be an $m \times m$ variable matrix. Then by classical representation theory [8] $\det(Y)$ is the unique nonzero polynomial, up to a constant multiple, in the variable entries y_{ij} of Y such that:

(D): (1) $\det(AY^*B) = \det(Y)$, for any $A, B \in SL_m(\mathbb{C})$, where $Y^* = Y$ or Y^t, and (2) $\det(\lambda Y) = \lambda^m \det(Y)$ for any $\lambda \in \mathbb{C}$. Thus $\det(Y)$ is characterized by its symmetries, and hence, is exceptional. We refer to this characteristic property of the determinant as property (D) henceforth.

Similarly, let X be an $n \times n$ variable matrix. Then by classical representation theory again [17] $\mathrm{perm}(X)$ is the unique nonzero polynomial, up to a constant multiple, in the variable entries x_{ij} of X such that for any diagonal or permutation matrices A, B,

(P): $\mathrm{perm}(AX^*B) = p(A)\mathrm{perm}(X)p(B)$,

where $X^* = X$ or X^t, and $p(A)$ is defined to be the product of diagonal entries, if A is diagonal, and one if A is a permutation matrix, $p(B)$ being similar. Thus $\mathrm{perm}(X)$ is also characterized by its symmetries, and hence, is exceptional. We refer to this characteristic property of the permanent as property (P) henceforth. In the proof of Lemma 4.1, only the property (P) is used. However, the property (D) is needed in the GCT approach to the permanent vs. determinant problem; see the overview [21].

For convenience, we now recall the elementary proof of property (P) [17], the proof of property (D) being similar. Let $f(X)$ be any polynomial with property (P). Letting A and B in (P) be diagonal matrices, it easily follows that $f(X)$ has the same total degree as $\mathrm{perm}(X)$, and also the same total degree (one) in the variables of any fixed row or column of X. This means that each monomial of $f(X)$ contains precisely one variable (with degree one) from each row and column of X. Thus it corresponds to a permutation of n symbols. Furthermore, letting A and B in (P) be permutation matrices, it follows that the coefficients of all monomials are the same. Hence $f(X)$ is a constant multiple of $\mathrm{perm}(X)$. This proves property (P).

5.2 Arithmetic P vs. NP problem

The function $E(X)$ (cf. Section 2) which plays the role of the permanent in the P vs. NP problem is also characterized by its symmetries (Theorem 5.1).

To state the result, we follow the same notation as in Section 2.2. Let K be the wreath product of the symmetric group S_k on k letters and the alternating group A_m on m letters. It acts on X by permuting its columns in the obvious way. We call X_{σ_0}, where $\sigma_0(i) = 1$ for all i, the *primary submatrix* of X, and $\det_{\sigma_0}(X) = \det(X_{\sigma_0})$ the primary minor of X.

The following is a strengthening of Proposition 7.2 in GCT1.

Theorem 5.1 *Let the base field F be of characteristic zero, say $\mathbb{Q}$ or $\mathbb{C}$. Then:*

(a) $E(X)$ is the only nonzero polynomial, up to a constant multiple, in the variable entries of X such that

(E)*:*

E1: for any $A \in GL_n(\mathbb{C})$ and any $B \in K$, $E(AXB) = (\det(A))^{km} E(X)$.

E2: (1) $E(X) = 0$ for any X with singular primary minor, or less stringently, (2) $E(X) = 0$ for any X whose primary minor has a unit $(n-1)\times(n-1)$ matrix as its top-left $(n-1)\times(n-1)$ minor and zeros in the bottom row.

(b) Let $e(X)$ be any integral nonzero polynomial satisfying E2 and the following variant of E1:

E1': for any $A \in SL_n(\mathbb{C})$ and any $B \in K$, $e(AXB) = e(X)$.

Then $e(X)$ can be written as $E(X)(\sum_\alpha a(\alpha)g(\alpha))$, $a(\alpha) \in \mathbb{C}$, where α ranges over monomials in the $m \times m$ minors of X, and

$$g(\alpha) = \sum_{B \in K} \alpha(XB).$$

We refer to the characterization of $E(X)$ in characteristic zero given by this result as property (E) henceforth.

Proof:

(a) Let $f(X)$ be any polynomial over $\mathbb{Q}$ or $\mathbb{C}$ with property (E). It is easy to see that E1 and E2 (2) together imply E2 (1). Hence, let us assume that $f(X)$ has properties E1 and E2 (1).

By E2 (1), $f(X) = 0$ if the primary minor of X is singular. Hence it easily follows from Hilbert's Nullstellansatz [30] that $f(X)$ is divisible by $\det(X_{\sigma_0})$, where X_{σ_0} denotes the primary $m \times m$ minor of X. Specifically, let $\mathcal{X}$ be the variety consisting of X's with singular primary minors. It is the zero set of the polynomial $\det(X_{\sigma_0})$. By E2 (1), $f(X)$ vanishes on $\mathcal{X}$. Hence, it follows from Hilbert's Nullstellansatz that $f(X)^r$, for some positive integer r, is divisible by $\det(X_{\sigma_0})$. Since $\det(X_{\sigma_0})$ is irreducible, it follows that $\det(X_{\sigma_0})$ divides $f(X)$.

Remark: The above special case of Nullstellansatz has an elementary proof. Specifically, let $\tilde{X} \in \mathcal{X}$ be a "generic" matrix with singular primary minor. Here generic means all entries of $\tilde{X}$ are algebraically independent except (say) the top-left, which is a rational function of the remaining entries of $\tilde{X}$ in such a way that the determinant of the primary minor of $\tilde{X}$ is zero. Then since $f(X)$ vanishes on $\tilde{X}$ and the determinant is irreducible, it is easy to show that $\det(X_{\sigma_0})$ divides $f(X)$.

Since, by E1, $f(XB) = f(X)$ for every $B \in K$, it now follows that $f(X)$ is divisible by $\det_\sigma(X)$, for every σ. That is, $f(X)$ is divisible by $E(X)$. It follows from E1, by letting $A = \lambda I \in GL_n(\mathbb{C})$, that $f(\lambda X) = \lambda^d f(X)$, for any $\lambda \in \mathbb{C}$, where $d = mk^m$ is the degree of $E(X)$. This means $f(X)$ is a homogeneous polynomial of the same degree as $E(X)$ and is divisible by $E(X)$. Hence, it is a constant multiple of $E(X)$. This proves (a).

(b) Now suppose that $e(X)$ is any nonzero polynomial satisfying E1' and E2. It follows as above that $e(X)$ is divisible by $E(X)$. By E1', $e(AX) = e(X)$ for any $A \in SL_n(\mathbb{C})$. Hence, by the first fundamental theorem of invariant theory [9, 37], $e(X)$ can be written as a polynomial in the $m \times m$ minors of X. Since $e(XB) = e(X)$ for any $B \in K$ and $E(X)$ divides $e(X)$, (b) follows. Q.E.D.

6 The stronger form of the $NP \not\subseteq P/poly$ conjecture

Before turning to the proof of the flip lemma and theorems, we prove in this section Proposition 2.2 following the same notation as in Conjecture 2.1.

Let $\sigma : F_p \to F_p$ be the Frobenius automorphism $x \to x^q$. For $x \in F_p$, let

$$\text{trace}(x) = \sum_{i=0}^{l-1} \sigma^i(x) = \sum_{i=0}^{l-1} x^{q^i}$$

denote its trace. It is known (Theorem 5.2 in chapter 6 in [15]) that the bilinear form trace(xy), $x, y \in F_p$, is nondegenerate. Fix a basis $B = \{b_i\}$, $0 \leq i \leq l-1$, of F_p over F_q. Let $\{b_i^*\}$ denote its dual basis with respect to the trace form. For any $x \in F_p$, let x_i's denote its coefficients in the basis B. Then $x_i = \text{trace}(b_i^* x)$. Hence, for any fixed i, $x_i \in F_q$ can be computed by an arithmetic F_p-circuit (with input x) of $O(l^2) = \text{poly}(n)$ size. Furthermore, since $q = \text{poly}(n)$, a bit representation of x_i can be computed by an F_q-circuit of poly(n) size using Lagrange interpolation. Thus, given $x \in F_p$, all bits of all x_i's can be computed by an arithmetic F_p-circuit of poly(n) size.

Now let $e(X) = E(X)^{p-1}$. Then $e(X)$ is 1 iff $E(X)$ is nonzero, and it is zero otherwise. Thus $e(X)$ is a boolean function that belongs to co-NP. So to prove the usual nonuniform $P \neq NP$ conjecture over the boolean field, it suffices to show that $e(X)$ can not be computed by a boolean circuit of poly(n) size, when the input to the circuit consists of the bits of the coefficients of X_{ij} (the entries of X) with respect to the basis B. Suppose to the contrary that such a circuit C exists. Then using C we can construct an arithmetic circuit C' over F_p of polynomial size computing $e(X)$. Specifically, we compute the bits of the coefficients of X_{ij} with respect to the basis B by small circuits as above and then feed these bits to C. By Conjecture 2.1 such a small C' for computing $e(X) = E(X)^{p-1}$ cannot exist. A contradiction. This proves Proposition 2.2.

7 Flip in the weak arithmetic setting

In this section we prove Lemma 4.1 and Theorem 4.2.

7.1 Proof of Lemma 4.1

Proposition 7.1 *[13] The problem of deciding if a given arithmetic circuit C over $\mathbb{Z}$ computes the permanent belongs to $co-RP$.*

Original proof: We first recall a proof from [13] and then give a new proof based on the property (P) that is crucially needed for proving F1 (b).

Given a circuit $C = C_n$ that is supposed to compute perm(X), dim$(X) = n$, we get a circuit C_i, $1 \leq i \leq n$, for computing perm(Y), dim$(Y) = i$, by putting Y in the lower right corner of X, specializing the remaining diagonal entries of X to 1, all others remaining entries to zero, and evaluating C_n on this X.

Let $C_i(Y)$ denote the value computed by C_i on input Y. Then C_n computes $\mathrm{perm}(X)$ if and only if for all $1 < i \le n$

$$C_i(Y) = \sum_{j=1}^{i} y_{1,j} C_{i-1}(Y_j), \tag{1}$$

where Y is an $i \times i$ variable matrix with variables $y_{k,l}$, and Y_j the j-th minor of Y along the first row, and

$$C_1(y) = y. \tag{2}$$

This is the usual downward self reducibility of the permanent. Testing if C_i's satisfy (1) and (2) is an arithmetic circuit (polynomial) identity testing problem (over $\mathbb{Z}$), which belongs to co-RP [11].

New GCT proof: By the property (P), $C(X) = \mathrm{perm}(X)$, up to a nonzero constant multiple, if and only if

$$C(X) \neq 0, \tag{3}$$

identically as a polynomial,

$$C(e_i X) = C(X) \text{ and } C(X e_i) = C(X), \quad \text{for all } i < n, \tag{4}$$

where e_i denotes an elementary permutation matrix (which permutes the ith and $(i+1)$-st positions), and

$$C(\mu X) = p(\mu) C(X) \text{ and } C(X\mu) = p(\mu) C(X), \tag{5}$$

where μ denotes a diagonal matrix, and $p(\mu)$ is the product of its diagonal entries.

Testing if $C(X)$ satisfies (3)–(5) is again an arithmetic circuit identity testing problem over $\mathbb{Z}$, which belongs to co-RP. Q.E.D.

7.1.1 Proof of Lemma 4.1 (1)

Now consider the second (new) co-RP algorithm in the proof above to test if C computes $\mathrm{perm}(X)$. This algorithm works in expected time $\le m' = m^c$, where $c > 1$ is some fixed constant. Assuming E does not have subexponential size circuits, it can be derandomized as follows. Article [12] gives, under this assumption, a $\mathrm{poly}(n, m)$ time computable pseudorandom generator g that takes a random seed of $l = O(\log m)$ bit size and produces a pseudorandom sequence of length m^c that fools any small circuit of bit size $\le m^c$. Consider the computational circuit corresponding to the above co-RP algorithm for testing if C computes $\mathrm{perm}(X)$. Feeding the pseudorandom sequence generated by g to this circuit in place of the random bits, cycling over all $\mathrm{poly}(m)$ possible seeds, and then taking a majority vote, we get a $\mathrm{poly}(n, m)$ time algorithm A for testing if C computes $\mathrm{perm}(X)$. (For this argument, we only need derandomization of polynomial identity testing, instead of the strong assumption that E does not have subexponential size circuits; cf. Section 7.4 for further discussion.)

A crucial property of A is that it is *nonadaptive*. This means the queries generated during its execution do not depend on C at all. Here a query specifies an X on which (3) is tested, or an X and i on which the equation (4) is tested, or a μ and an X on which (5) is tested. Let $Q_{n,m}$ denote the set of $\mathrm{poly}(n,m)$ queries generated in A when the input to A is a circuit C of bit size $\leq m$. Nonadaptiveness means $Q_{n,m}$ depends only on n and m but not on C at all.

Assuming that $\mathrm{perm}(X)$ cannot be computed by a circuit of bit size $m = \mathrm{poly}(n)$, it follows that, when $m = \mathrm{poly}(n)$, then for *every* C of bit size $\leq m$, $Q_{n,m}$ contains a query on which an algebraic identity test based on (3), (4) or (5) fails for that C. Let $S_{n,m}$ be the set of all inputs X's on which C is evaluated during the testing of all queries in $Q_{n,m}$. Specifically, fix a query q in $Q_{n,m}$. Suppose this query requires testing of the first equation in (4) for some fixed $i < n$ and $X = X_q$ for some input X_q, the argument for the second equation being similar. Then during the course of testing this equation for this query, we evaluate C on X_q as well $e_i X_q$ (The evaluation in the co-RP algorithm [11] for algebraic identity testing works modulo a large enough prime to keep the bit sizes under control. But this makes no difference in the argument that follows.) So there are two values of X (namely X_q and $e_i X_q$) on which C is evaluated during the testing of this query. Let $S_q = \{X_q, e_i X_q\}$ and add both elements in S_q to $S_{n,m}$ for this query. If the query q requires testing of the first (say) equation in (5) on some fixed value μ_q of μ and X_q of X, then we let $S_q = \{X_q, \mu_q X_q\}$, add both X_q and $\mu_q X_q$ to $S_{n,m}$. If the query requires testing of (3) on some X_q, we let $S_q = \{X_q\}$, and add X_q to $S_{n,m}$. Thus $S_{n,m} = \cup_q S_q$ contains a set of $\mathrm{poly}(n,m)$ $n \times n$ matrices. Because $Q_{n,m}$ contains, for *every* C of bit size $\leq m$, a query on which the associated algebraic identity test fails, it follows that $S_{n,m}$ also contains, for *every* circuit C of bit size $\leq m$, a matrix X_C on which $C(X_C) \neq \mathrm{perm}(X_C)$. Thus $S_{n,m}$ is a small global obstruction set against all circuits of bit size $\leq m$. Furthermore, using the algorithm A, we can compute $S_{n,m}$ in $\mathrm{poly}(n,m)$ time. This proves statement (1) of Lemma 4.1.

7.1.2 Proof of Lemma 4.1 (2)

Now we turn to the construction of the obstruction family $\tilde{\mathcal{O}} = \tilde{\mathcal{O}}_{n,m}$ as needed in the statement (2) of Lemma 4.1. Let $m' = m^c$ be the bound on the running time of A as above. Let $l = b \log m$, for a large enough constant $b > c$. For small m (i.e. $m = \mathrm{poly}(n)$), let $\mathcal{O}_{n,m}$ be the set of all $(\log m', a \log m)$-designs within the set $\{1, \ldots, l\}$, for a large enough constant $c < a < b$. Here by a (k,r)-design within $\{1, \ldots, l\}$, we mean [32] a collection of sets $\{T_1, \ldots, T_{m'}\}$, $T_i \subseteq \{1, \ldots, l\}$, such that (1) for all i, $|T_i| = r$, and (2) for all $i \neq j$, $|T_i \cap T_j| \leq k$. Each such design s can be specified by an $m' \times l$ boolean adjacency matrix whose i-th row specifies T_i (by letting its j-th entry be one if T_i contains j and zero otherwise). The bitlength $\langle s \rangle$ of this specification is $O(m' \log m) = O(\mathrm{poly}(n,m))$. This s in $\mathcal{O}_{n,m}$ is short if m is small. It is easy to see (from the proof of Lemma 2.6 in [32]) that the total number of such designs is $\geq 2^{\Omega(m'l)} = 2^{\Omega(m' \log m)}$.

We now verify that this construction satisfies F0-F4. For the proof of F2 we will need a complexity theoretic conjecture.

F0: This is clear by the preceding remark on the number of designs.

F1 (a): It follows from the results in [12] that, for each design $s \in \mathcal{O}_{n,m}$, there exists a $\mathrm{poly}(m)$-time computable pseudo-random generator $g(s)$ that takes a random seed of $l = O(\log m)$ bit length and produces a pseudorandom sequence of bit length $m' = m^c$ that fools any circuit of bit size $\leq m'$. When m is small (and thus s is short), using this pseudo-random generator $g(s)$

in place of the pseudo-random generator g above, we can compute a small global obstruction set $S_{n,m}(s)$ in poly(n, m) time, so also the associated set $Q_{n,m}(s)$ of queries. This proves F1 (a).

F1 (b): Given $n, m = \text{poly}(n)$, a short s, and a circuit C, $Q_{n,m}(s)$ is guaranteed to contain a query q on which C fails, and this query q can be computed in poly$(n, m, \langle s \rangle) =$ poly(n, m) time. Let $S_q(s)$ be the associated set of X's on which C is evaluated during the testing of this query. The size of $S_q(s) \leq 2$. Let $S_{n,m,C}(s) = S_q(s)$. Clearly it too can be computed in poly(n, m) time.

F3: Given a design $s \in \mathcal{O}_{n,m}$ specified as an $m' \times l$ adjacency matrix, whether it is a valid $(\log m', a \log m)$ design within $\{1, \ldots, l\}$, $l = b \log m$, can be clearly verified in poly(n, m) time.

F4: Lemma 2.6 in [32] gives an algorithm to compute one such valid design in poly(n, m) time.

F2: This follows from:

Conjecture 7.2 *The pseudorandom generator $g(s)$ given by [12] under the assumption that E does not have subexponential size circuits has the following additional property: for a fixed constant c, and large enough constants $a > c$ and $b > a$, $\mathcal{O}_{n,m}$ contains $2^{\Omega(m)}$ mutually disjoint s's (as we would expect if s's are sufficiently (pseudo)-random). Here we say that s, s' are mutually disjoint if $S_{n,m}(s)$ and $S_{n,m}(s')$ are mutually disjoint.*

This is a slightly strengthened version of the following conjecture that only depends on the complexity class E, and not on the permanent vs. determinant problem or the property (P).

Let $R_m(s)$ denote the set of pseudorandom sequences of length m^c produced by $g(s)$ as the seed ranges over all possible bit-strings of length $l = b \log m$.

Conjecture 7.3 *The pseudorandom generator $g(s)$ given by [12] under the assumption that E does not have subexponential size circuits has the following additional property: for a fixed constant c, and large enough constants $a > c$ and $b > a$, the collection $\{R_m(s)\}$, $s \in \mathcal{O}_{n,m}$, contains at least $\Omega(2^{\Omega(m)})$ mutually disjoint sets.*

Each string in $R_m(s)$ contributes poly(n) X's to $S_{n,m}(s)$, instead of just one, and hence disjointness of $S_{n,m}(s)$'s in Conjecture 7.2 is a bit stronger than disjointness of $R_m(s)$'s above. Conjectures 7.2 and 7.3 stipulate pseudo-randomness of the generator in [12] with respect to a new measure in addition to the usual one used there.

7.1.3 Proof of Lemma 4.1 (3)

This is similar to that of Lemma 4.1 (1) and (2).

This finishes the proof of Lemma 4.1.

7.2 Characterization by symmetries vs. self reducibility

It is illuminating to consider what happens if we use in the preceding proof the first (original) co-RP algorithm in the proof of Proposition 7.1 instead of the second (new) one as we did. Then we cannot prove F1 (b). Because each query to test (1) in Proposition 7.1 requires $O(n)$

evaluations of the circuit C. Hence the size of $S_{n,m,C}(s)$ in this case would be $O(n)$ and not $O(1)$ as needed in F1 (b). Thus the new co-RP algorithm is crucial to bring down the size of $S_{n,m,C}(s)$ from $O(n)$ to $O(1)$.

In the context of the arithmetic P vs. NP problem that we turn to next, characterization by symmetries is even more important. Because in this context we do not know how to use downward self-reducibility to prove any *any* flip theses. Specifically, the best result based on downward self reducibility for the usual nonuniform P vs. NP problem is the one in [3], which as we already discussed after Theorem 4.2, does not efficiently yield a global obstruction set against all circuits (i.e., cannot even satisfy F1 (a)). This is akin to a similar phenomenon that has already been observed in complexity theory: namely, we know how to use random self reducibility to reduce worst case hardness to average case hardness in the context of the $\#P$ vs. P problem, but not in the context of the P vs. NP problem, and indeed, there is compelling evidence [4, 6] that the usual reduction strategies based on self reducibility would not work in the context of the P vs. NP problem.

7.3 Proof of Theorem 4.2

For these reasons, Theorem 4.2 proved in this section is the main result in the weak arithmetic setting.

The following is the analogue of Proposition 7.1 in this case.

Proposition 7.4 *The problem of deciding if a given arithmetic circuit C over $\mathbb{Z}$ computes $E(X)$ belongs to $co-RP$.*

Proof: For any $y \in \mathbb{C}$, and $i \neq j$, let $e_{ij}(y)$ denote an elementary $n \times n$ matrix with 1's on the diagonal, y in the (i,j)-th place, and zeroes everywhere else. By the proof of Guassian elimination, any matrix in $GL_n(\mathbb{C})$ can be written as a product of elementary matrices, where by an elementary matrix we mean a matrix of the form $e_{ij}(y)$, or a diagonal matrix, or an elementary permutation matrix (that swaps some fixed two rows or columns). The total number of types of elementary matrices is clearly $O(n^2)$. Fix an explicit set $\{f_j\}$ of generators for the group K (defined before the statement of Theorem 5.1) so that the total bit length of their description is $O(\text{poly}(n))$.

By property (E) as per Theorem 5.1, $C(X) = E(X)$ up to a nonzero constant multiple if and only if

$$E(X) \neq 0 \tag{6}$$

identically as a polynomial,

$$C(eX) = C(X), \tag{7}$$

for any elementary matrix e,

$$C(X) = C(Xf_j), \quad \text{for all} \quad j, \tag{8}$$

and

$$C(X) = 0 \tag{9}$$

for any X such that, X_i^1, for each $i < m$, is a vector with 1 in the i-th location and zero everywhere else, and the m-entry of X_m^1 is zero. This last condition tests the property E2 (2).

Testing if $C(X)$ satisfies (6)–(9) is an arithmetic circuit (polynomial) identity testing problem over $\mathbb{Z}$, which belongs to co-RP. Specifically, to test (6) we choose X randomly. We need to test (7) separately for each type of e. If e is of the type $e_{ij}(y)$, we choose y randomly and test (7) by choosing X randomly. Similarly if e is diagonal. If e is an elementary permutation matrix, we just have to choose X randomly. Similarly for (8). For testing (9), we have choose X randomly subject to the condition on X specified there. Q.E.D.

Testing (7) for a given elementary e and a given X requires only $O(1)$ evaluations of the circuit C, and similarly for (6), (8) and (9), just as in the case of (3), (4) or (5). The rest of the proof of Theorem 4.2 is now like that of Lemma 4.1 using Proposition 7.4 instead of Proposition 7.1.

For the proof of F2, the following conjecture plays the role of Conjecture 7.2.

Conjecture 7.5 *Analogue of Conjecture 7.2 holds assuming that $S_{n,m}(s)$ is defined using (derandomization) of the algorithm in Proposition 7.4 instead of the one in Proposition 7.1.*

This finishes the proof of Theorem 4.2.

7.4 Derandomization of black box polynomial identity testing

The proofs of Lemma 4.1 and Theorem 4.2 above also go through if instead of assumming that E does not have subexponential size circuits, we assume instead that black box polynomial identity testing [1, 13] can be derandomized. By this we mean that there exists a family $\mathcal{H} = \cup_{n,m} H_{n,m}$ such that:

1. *Short:* Each element h of $H_{n,m}$ is a short hitting set [1] against all arithmetic circuits over $X = (x_1, \ldots, x_n)$ of bit size $\leq m$. By a short hitting set h we mean a set $\{X_1, \ldots, X_l\}$, $l = \text{poly}(m)$, of inputs of total bit size $O(\text{poly}(n, m))$ such that for every circuit C of total bit size $\leq m$ that computes a nonzero polynomial, h contains an input $X_C = X_i$, $i \leq l$, such that $C(X_C) \neq 0$.

2. *Rich:* $H_{n,m}$ contains at least $2^{\Omega(m)}$ pairwise disjoint hitting sets.

3. *Easy to verify:* Given n, m and h, whether $h \in H_{n,m}$ can be verified in $\text{poly}(n, m, \langle h \rangle)$ time, where $\langle h \rangle$ denotes the bit length of h.

4. *Easy to construct:* Given n and m, a short $h \in H_{n,m}$ can be constructed in $\text{poly}(n, m)$ time.

The proof of Lemma 4.1 shows that this derandomization hypothesis holds if E does not have subexponential size circuits. We leave the details of reworking the proofs of Lemma 4.1 and Theorem 4.2 with this less stringent derandomization hypothesis, instead of the assumption about E, to the reader. No additional conjectures such as Conjecture 7.2 or 7.5 are needed in this case.

Derandomization of black box polynomial identity testing is roughly equivalent to proving subexponential arithmetic circuit size lower bounds for multilinear functions in E; cf. Section 7.3 in [13] and Section 5 in [1]. The notion of derandomization here is a bit stronger than that in [1, 13]. But the proofs there can be extended to this stronger setting easily.

7.5 Proofs of Lemma 4.4 and Theorem 4.5

This follows by uniformizing the proofs of Lemma 4.1 and Theorem 4.2. We omit the details.

8 Flip in the arithmetic setting

In this section we prove Theorem 4.3.

8.1 Strong derandomization hypothesis

We begin by specifying the strong derandomization hypothesis mentioned in the statement of Theorem 4.3. It is a natural generalization of the derandomization hypothesis in the weak arithmetic setting described in Section 7.4.

Let C be an arithmetic circuit over $X = (x_1, \ldots, x_n)$ of size $\leq m$. Let $S = [1, 2^{m^2}]$ be the set of integers between 1 and 2^{m^2} (say). Since the degree of $C(X)$ is $\leq 2^m$, by the standard lemma [34], the result of evaluating C is nonzero with a high probability if X is assigned a random element in S^n. It is critical here that the size of S does not depend on the bitsize of the constants in C, since we are allowing arbitrary constants from $\mathbb{C}$ in C. Indeed, constants may not even have specifications of finite bitlength if they are transcendental. Now we have a natural randomized polynomial time algorithm in the complex-RAM model for deciding if $C(X)$ is identically zero: (1) pick a random element in S^n, (2) evaluate $C(X)$, (3) say no if $C(X)$ is not zero, and (4) yes otherwise. In the complex-RAM model each memory location contains a complex number, and each arithmetic operation $(+, -, *)$ is unit-cost. This is a black-box algorithm in the sense that it treats the circuit C as a black-box subroutine.

The derandomization hypothesis in the arithmetic setting is that this black box polynomial identity testing can be derandomized. By this we mean that there exists a family $\mathcal{H} = \cup_{n,m} H_{n,m}$ such that:

1. *Short:* Each element h of $H_{n,m}$ is a short hitting set [1] against all arithmetic circuits over $X = (x_1, \ldots, x_n)$ of size (rather than bit size) $\leq m$. By a short hitting set h we mean a set $\{X_1, \ldots, X_l\}$, $l = \text{poly}(n, m)$, of inputs of total bit size $O(\text{poly}(n, m))$ such that for every circuit C of size $\leq m$ that computes a nonzero polynomial, h contains an input $X_C = X_i$, $i \leq l$, such that $C(X_C) \neq 0$.
2. *Rich:* $H_{n,m}$ contains at least $2^{\Omega(m)}$ pairwise disjoint hitting sets.
3. *Easy to verify:* Given n, m and h, whether $h \in H_{n,m}$ can be verified in $\text{poly}(n, m, \langle h \rangle)$ time, where $\langle h \rangle$ denotes the bit length of h.

4. *Easy to construct:* Given n and m, a short $h \in H_{n,m}$ can be constructed in poly(n,m) time.

Lemma 8.1 *The arithmetic derandomization hypothesis above holds assuming that E does not have subexponential size circuits, or less stringently, that analogous derandomization hypothesis holds over F_p, with the bitlength $\langle p \rangle = O(m^2)$, say.*

The derandomization hypothesis over F_p is just like the arithmetic derandomization hypothesis above with the arithmetic circuits of size $\leq m$ replaced by circuits over F_p of size $\leq m$, and requiring each input in the hitting set to be over F_p instead of $\mathbb{Z}$.

Proof: From the proof of Lemma 4.1 it follows (after appropriate modifications) that the derandomization hypothesis over F_p holds assuming that E does not have subexponential size circuits. Since each arithmetic circuit over $\mathbb{Z}$ corresponds to a circuit over F_p obtained by reducing it modulo F_p, the derandomization hypothesis over F_p implies the arithmetic derandomization hypothesis over $\mathbb{Z}$. Q.E.D.

The strong arithmetic derandomization hypothesis in the strong arithmetic setting is obtained by letting $C(X)$ in the arithmetic hypothesis above be any function that can be approximated infinitesimally closely by circuits of size $\leq m$. Thus the hitting set is now against all functions that can be approximated infinitesimally closely by circuits of size $\leq m$.

Arithmetic and strong arithmetic derandomization hypotheses in the context of the permanent vs. determinant problem are defined similarly (by replacing circuit identity testing by determinant identity testing).

8.2 Proof of Theorem 4.3

We now describe how to extend the proof of Theorem 4.2 to that of Theorem 4.3. We only consider Theorem 4.3 (a), since (b) is very similar.

The conditions F0-4 in Theorem 4.3 can be proved just like those in Theorem 4.2 in the weak arithmetic setting, letting the (strong) arithmetic hardness conjecture play the role of the weak arithmetic hardness conjecture, and letting the (strong) derandomization hypothesis above play the role of the weak derandomization hypothesis in Section 7.4.

What remains to prove then is the property G. We turn to this next.

We follow the terminology in the statement of the property G in Section 3.2. Thus, given $s \in \mathcal{O}_{n,m}$, $S_{n,m}(s) = \{X_1, \ldots, X_l\}$, $l = \text{poly}(n,m)$, denotes the small glbal obstruction set as in F1 (a). The space V is the space of polynomial functions in X of degree $\leq d = 2^m$, and $\Sigma_{n,m}$ is the set of the functions in V that can be computed by arithmetic circuits of size $\leq m$.

Let z be an additional homogenizing variable. Given any $g(X) \in V$, let $g'(z,X)$ denote the homogeneous polynomial of degree precisely $d = 2^m$ obtained from $g(X)$ by homogenizing it using z. Let V' denote the space of homogenizations of the polynomials in V. Let $\Sigma'_{n,m} \subseteq V'$ denote the set of all constant multiples of homogenizations of all polynomials in $\Sigma_{n,m}$. This set is homogeneous; i.e. if $g'(z,X) \in \Sigma'_{n,m}$, then $ag'(z,X) \in \Sigma'_{n,m}$ for all $a \in \mathbb{C}$. Let $\text{perm}'(z,X) = z^{d-n}\text{perm}(X) \in V'$ be the homogenization of perm(X). Let $\psi' = \psi'_s : V' \to \mathbb{C}^l$ denote the homogeneous linear map such that for any $g'(z,X) \in V'$, and any $i \leq l$,

$$\psi'_s(g'(z,X))_i = g'(1,X_i).$$

In other words, $\psi'_s(g'(z,X))$ is simply the l-tuple of evaluations of $g'(z,X)$ at various X_i's, letting $z=1$, and $\psi'_s(g'(z,X))_i$ denotes the i-th entry in this tuple.

It is easy to show that any $g'(z,X) \in \Sigma'_{n,m}$ can be computed by an arithmetic circuit over $\mathbb{C}$ with input z and X and of size $\leq m' = bm^2$ for some large enough constant b. (The proof proceeds by induction on the depth of the circuit computing $g'(z,X)$.) Hence it follows from the strong arithmetic hardness conjecture for $\mathrm{perm}(X)$ that $\mathrm{perm}'(z,X)$ does not belong to the closure $\bar{\Sigma}'_{n,m}$ of $\bar{\Sigma}_{n,m}$ in the complex topology. Assuming the strong derandomization hypothesis (cf. Section 8.1), it follows as in the proof of F1 in the strong arithmetic setting above, that, for any $s \in \mathcal{O}_{n,m'}$, $S_{n,m'}(s)$ is also a global obstruction set against all functions in $\bar{\Sigma}'_{n,m}$. Specifically, this means that $\psi'_s(\mathrm{perm}'(z,X)) \notin \psi'_s(\bar{\Sigma}'_{n,m})$. Replacing $\mathcal{O}_{n,m}$ by $\mathcal{O}_{n,m'}$ in the obstruction family $\mathcal{O}$, we will assume, without of loss of generality, that, for any $s \in \mathcal{O}_{n,m}$, $S_{n,m}(s)$ is a global obstruction set against all functions in $\bar{\Sigma}'_{n,m}$. This means

$$\psi'_s(\mathrm{perm}'(z,X)) \notin \psi'_s(\bar{\Sigma}'_{n,m}) \tag{10}$$

for any $s \in \mathcal{O}_{n,m}$.

Let $P(V')$ be the projective space of lines in V' through the origin. Let $P(\mathbb{C}^l)$ be the similar projective space associated with $\mathbb{C}^l$. Let $P(\bar{\Sigma}'_{n,m}) \subseteq P(V)$ denote the projective set associated with $\bar{\Sigma}'_{n,m}$. We can assume, without loss of generality that, for any function $g'(z,X)$ in $\bar{\Sigma}'_{n,m}$, $S_{n,m}(s)$ contains a matrix X_C such that $g'(1,X_C) \neq 0$; i.e., $\psi'_s(g'(z,X))$ is not an identically zero tuple. This is because the test for the property (P) also includes the test that the function under consideration is not identically zero (cf. eq.(3)), and $S_{n,m}(s)$ is constructed on the basis of the property (P). Thus ψ'_s gives a well defined map from $P(\bar{\Sigma}'_{n,m})$ to $P(\mathbb{C}^l)$. We denote this map by $\hat{\psi}'_s$. We can also assume without loss of generality that each $S_{n,m}(s)$ contains an identity matrix. Since the permanent of the identity matrix is one, this means $\psi'_s(\mathrm{perm}'(z,X))$ is also not an identically zero tuple. We denote the point in $P(\mathbb{C}^l)$ corresponding to $\psi'_s(\mathrm{perm}'(z,X))$ by $\hat{\psi}'_s(\mathrm{perm}'(z,X))$. Thus, by eq.(10),

$$\hat{\psi}'_s(\mathrm{perm}'(z,X)) \notin \hat{\psi}'_s(P(\bar{\Sigma}'_{n,m})) \subseteq P(\mathbb{C}^l) \tag{11}$$

for any $s \in \mathcal{O}_{n,m}$.

To prove the property G for the permanent function, it suffices to show that $\psi'_s(\mathrm{perm}'(z,X))$ does not belong to the closure of $\psi'_s(\bar{\Sigma}'_{n,m})$ in the complex topology. This is equivalent to showing that $\hat{\psi}'_s(\mathrm{perm}'(z,X))$ does not belong to the closure of $\hat{\psi}'_s(P(\bar{\Sigma}'_{n,m}))$ in the complex topology. By eq.(11), this follows from the following.

Lemma 8.2 *The set $\hat{\psi}'_s(P(\bar{\Sigma}'_{n,m})) \subseteq P(\mathbb{C}^l)$ is already closed in the complex topology.*

Fix n,m and $s \in \mathcal{O}_{n,m}$. For simplicity, we drop the subscripts s,n and m. Thus we denote $\hat{\psi}'_s$ by $\hat{\psi}'$, $\Sigma'_{n,m}$ by Σ', and $\bar{\Sigma}'_{n,m}$ by $\bar{\Sigma}'$.

To prove lemma 8.2, we need the following lemma.

Lemma 8.3 *The set $\bar{\Sigma}' \subseteq V'$ is an algebraic variety (possibly reducible); i.e. the zero set of finitely many polynomials in the coordinates of V'.*

Proof: This follows from the following two facts from classical algebraic geometry:

(1) The set $\Sigma \subseteq V$ and hence the set $\Sigma' \subseteq V'$ is a constructible set. (A set is called constructible (cf. Definition 2.30 in [30]) if it can be expressed as a disjoint union $T_1 \cup \cdots \cup T_k$, where each $T_k = T_k' - T_k''$ for some algebraic variety T_k' and its subvariety $T_k'' \subseteq T_k'$.)

This can be proved as follows. Fix an uninstantiated circuit D of size $\leq m$. By an uninstantiated circuit, we mean the nodes of D are labelled with the operators $+, -$ and $*$, and the leaves are labelled by either the variables x_i's, or constant parameters $a_1, \ldots, a_j$, for some $j < m$. Clearly there are only finitely many uninstantiated circuits for given m. Fix any such D. Let $\Sigma_D \subseteq V$ be the set of all functions that can be computed by some instantiation of D; i.e., by assigning specific complex values to the constant parameters $a_1, \ldots, a_j$. Clearly, $\Sigma = \cup \Sigma_D$. So it suffices to show that Σ_D is constructible. With D, we can associate an affine algebraic variety as follows. Associate a new variable y_u with every internal node u of D. (The leaves of D are already associated with either variables x_i's or constant parameters a_r's). Say the internal node u is $*$, and u_1 and u_2 are its children, possibly leaves. Then corresponding to u, we have an equation $y_u = y_{u_1} * y_{u_2}$. Let Π_D denote the affine variety defined by all the equations associated with the internal nodes. Then Σ_D is the projection of Π_D into V. (This corresponds to elimination of all variables for the internal nodes and the parameters $a_1, \ldots, a_j$). Now (1) follows from the fact (cf. Proposition 2.31 in [30]) that the image of any affine variety under a regular (polynomial) map is a constructible set. It need not be closed. See Chapter 2C in [30] for the pathologies that can happen. This is the main problem that we have to deal with in the rest of the proof.

(2) The closure in the complex topology coincides with the closure in the Zariski topology (cf. Theorem 2.33 in [30]).

Specifically this implies the following. Since by (1), Σ' is a constructible set, its closure $\bar{\Sigma}'$ in the complex topology is an algebraic variety (possibly reducible–we do not require a variety to be reducible in what follows). Q.E.D.

Since Σ' is homogeneous, its closure $\bar{\Sigma}' \subseteq V'$ is also homogeneous. In conjunction with lemma 8.3, this means $\bar{\Sigma}'$ is a homogeneous algebraic subvariety of V'. Hence $P(\bar{\Sigma}')$ is a projective subvariety of $P(V')$. Consider the morphism $\hat{\psi}' = \hat{\psi}_s'$ from $P(\bar{\Sigma}')$ to $P(\mathbb{C}^l)$ defined earlier. To prove Lemma 8.2, it suffices to show that $\hat{\psi}'(P(\bar{\Sigma}'))$ is a projective subvariety of $P(\mathbb{C}^l)$. This follows from the fact that the image of a morphism from a projective variety to another projective variety is closed (cf. Corollary 14.2 in [5])–this is a consequence of the main theorem of elimination theory (cf. Theorem 14.1 in [5]). This proves Lemma 8.2.

Now the property G follows. This proves Theorem 4.3 (a).

9 Implication in algebraic geometry

The algebraic variety $\bar{\Sigma}'_{n,m}$ associated above with the class of functions computable by small arithmetic circuits is rather wild and hard to study. The article GCT1 associates another variety with this class of functions. It is called the *class variety* associated with the complexity class

P. Unlike $\bar{\Sigma}'_{n,m}$, it has a natural action of the general linear group $GL_{m^2}(\mathbb{C})$. This makes it possible to study it using the techniques of geometric invariant theory [31]. The article GCT1 also associates similar class varieties with other complexity classes, namely, NC, NP and $\#P$. Theorem 4.3 implies that a formidable explicit construction problem associated with these class varieties can be solved (in polynomial time) assuming the strong arithmetic hardness and derandomization hypotheses under consideration. To see this, one simply has to rephrase Theorem 4.3 in terms of these varieties. We do it in this section for the case of the strong arithmetic permanent vs. determinant problem, the other cases being similar.

Towards that end, we first recall the class varieties associated by GCT1 with the complexity classes NC and $\#P$. Let Y be an $m \times m$ variable matrix. We think of its entries, ordered say rowwise, as coordinates of $\mathcal{Y} = \mathbb{C}^r$, $r = m^2$. Let $V = \mathbb{C}[Y]_m$ be the space of homogeneous polynomials of degree m in the variable entries of Y. It is a representation of $G = GL(\mathcal{Y}) = GL_r(\mathbb{C})$ with the following action. Given any $\sigma \in G$, map a polynomial $g(Y) \in V$ to $g^\sigma(Y) = g(\sigma^{-1}(Y))$:

$$\sigma : g(Y) \longrightarrow g(\sigma^{-1}Y).$$

Here Y is thought of as an m^2-vector by straightening it rowwise.

Similarly, let X be an $n \times n$ variable matrix, whose entries we think of as coordinates of $\mathcal{X} = \mathbb{C}^{n^2}$ after ordering them rowwise. Let $W = \mathbb{C}[X]_n$ be the space of forms (homogeneous polynomials) of degree n in the entries of X. It is a representation of $H = GL(\mathcal{X}) = GL_{n^2}(\mathbb{C})$.

Let $P(V)$ be the projective space of V consisting of the lines in V through the origin. Let $P(W)$ be the projective space of W. Identify X with an $n \times n$ submatrix of Y, say, the bottom-right minor of Y, and let z be any variable entry of Y outside X. We use it as a homogenizing variable. Define an embedding $\phi : W \hookrightarrow V$ by mapping any polynomial $h(X) \in W$ to $h^\phi(Y) = z^{m-n}h(X)$. This also defines an embedding of $P(W)$ in $P(V)$, which we denote by ϕ again.

Let $g = \det(Y)$, thought of as a point in $P(V)$ (strictly speaking the line through $\det(Y)$ is a point in $P(V)$, but we ignore this distinction here). Similarly, let $h = \mathrm{perm}(X) \in P(W)$, and $f = h^\phi = \mathrm{perm}^\phi(Y) \in P(V)$.

Let

$$\begin{array}{lllll} \Delta_V[g,m] & = & \Delta_V[g] & = & \overline{Gg} \subseteq P(V), \\ \Delta_W[h,n] & = & \Delta_W[h] & = & \overline{Hh} \subseteq P(W), \\ \Delta_V[f,n,m] & = & \Delta_V[f] & = & \overline{Gf} \subseteq P(V), \end{array} \tag{12}$$

where $\overline{Gg}$ denotes the projective closure of the orbit Gg of g, and so on. Then, it follows from classical algebraic geometry as in the proof of Lemma 8.3 that $\Delta_V[g,m]$ and $\Delta_V[f,m,n]$ are projective varieties. Furthermore, it can be shown that they are projective G-varieties, i.e., varieties with a natural action of G induced by the action on the G-orbits. Similarly, $\Delta_W[h,n]$ is a projective H-variety. We call $\Delta[f,n,m]$ the *class variety* of the complexity class $\#P$ since the permanent is $\#P$-complete [36], and $\Delta[g,m]$ the *class variety* of the complexity class NC since the determinant belongs to NC and is almost complete [36].

It is easy to show (cf. Propositions 4.1 and 4.4 in [GCT1]) that if $h = \mathrm{perm}(X)$ can be expressed linearly as the determinant of an $m \times m$ matrix, $m > n$, then

$$\Delta_V[f] = \Delta_V[f,n,m] \subseteq \Delta_V[g,m] = \Delta_V[g], \tag{13}$$

and conversely, if $\Delta_V[f,n,m] \subseteq \Delta_V[g,m]$, then f can be approximated infinitesimally closely by a point in $P(V)$ of the form $\det(AY)$, $A \in G$, thinking of Y as an m^2-vector. The following conjecture is thus equivalent to the strong arithmetic permanent vs. determinant conjecture stated in Section 2.

Conjecture 9.1 *(Strong arithmetic form of the permanent vs. determinant conjecture) [GCT1] The point $f \in P(V)$ cannot be approximated infinitesimally closely as above if $m = poly(n)$, and more generally, $m = 2^{\log^a n}$ for any constant $a > 0$.*

Equivalently, if $m = poly(n)$, or more generally, $m = 2^{\log^a n}$, $a > 0$ fixed, $n \to \infty$, then $\Delta_V[f,n,m] \not\subseteq \Delta_V[g,m]$.

We now restate Theorem 4.3 for this equivalent form of the strong arithmetic permanent vs. determinant conjecture.

An obstruction $s \in \mathcal{O}_{n,m}$ will now be against all points (functions) in $\Delta_V[g,m]$. Specifically, the global obstruction set $S_{n,m}(s) = \{X_1, \ldots, X_l\}$, $l = \text{poly}(n,m)$, will now have the following property. Fix any homogeneous polynomial $p(Y)$ in V that belongs to $\Delta_V[g,m]$ (thinking of a homogeneous polynomial in V, by an abuse of notation, as a point in $P(V)$). Then there exists a counter example $X_i \in S_{n,m}(s)$ such that $p'(X_i) \neq \text{perm}(X_i)$, where $p'(X_i)$ is a polynomial obtained from $p(Y)$ by substituting zero for all variables in Y other than z and X, substituting 1 for z, and X_i for X. Equivalently, let $\psi = \psi_s : V \to \mathbb{C}^l$ be the homogeneous linear map that maps any homogeneous $p(Y) \in V$ to the point in $\mathbb{C}^l$ corresponding to the tuple $(p'(X_1,), \ldots, p'(X_l))$. As in the proof of Theorem 4.3 in Section 8.2, we can assume, without loss of generality, that ψ gives a well defined morphism from the projective variety $\Delta_V[g,m]$ to the projective variety $P(\mathbb{C}^l)$. We denote this morphism by $\hat{\psi} = \hat{\psi}_s$. Its image is $\hat{\psi}(\Delta_V[g,m]) \subseteq P(\mathbb{C}^l)$. We can also assume, as in the proof of Theorem 4.3 in Section 8.2, that $\psi(f) \in \mathbb{C}^l$ is not an identically zero tuple. Hence it defines a point in $P(\mathbb{C}^l)$, which we define by $\hat{\psi}(f)$. Then that $S_{n,m}(s)$ is a global obstruction set is equivalent to saying that $\hat{\psi}(f) \notin \hat{\psi}(\Delta_V[g,m])$. The notion of an explicit proof and F0-4 can now be formulated in this setting in the obvious manner; we omit the details. Note that, since $\hat{\psi}$ is a well defined morphism from the projective variety $\Delta_V[g,m]$ to the projective variety $P(\mathbb{C}^l)$, its image $\hat{\psi}(\Delta_V[g,m]) \subseteq P(\mathbb{C}^l)$ is already closed (projective subvariety) in $P(\mathbb{C}^l)$ by the main theorem of elimination theory (cf. Corollary 14.2 in [5]). Hence the property G follows from F0-4 in this setting immediately by the main theorem of elimination theory.

The following is a restatement of Theorem 4.3 in this setting.

Theorem 9.2 (Flip) *Assume Conjecture 9.1 and the strong arithmetic derandomization hypothesis (cf. Section 8.1). Then Conjecture 9.1 has an explicit proof satisfying F0-4 and G as above.*

More specifically, for any obstruction $s \in \mathcal{O}_{n,m}$, there is a linear map $\psi_s : V \to \mathbb{C}^l$ corresponding to the polynomial time computable global obstruction set $S_{n,m}(s)$ such that (1) it gives a well defined morphism $\hat{\psi}_s$ from $\Delta_V[g,m]$ to $P(\mathbb{C}^l)$, (2) $\hat{\psi}_s(\Delta_V[g,m])$ is a closed projective subvariety of $P(\mathbb{C}^l)$, and (3) $\hat{\psi}_s(f) \notin \hat{\psi}_s(\Delta_V[g,m])$.

Analogous result holds in the context of the strong arithmetic P vs. NP problem, letting the similar variety for the class P defined in GCT1 play the role of $\Delta_V[g,m]$ and letting the function $E(X)$ play the role of perm(X).

We call the linear map $\hat{\psi}_s$ in Theorem 9.2 an *explicit separator* between $\Delta_V[g, m]$ and $f = \text{perm}^\phi(Y)$. We call it explicit because, given s, its specification $S_{n,m}(s)$ can be computed in $O(\text{poly}(n, m))$ time. We call $l = \text{poly}(n, m)$ the *dimension* of $\hat{\psi}_s$. Thus Theorem 9.2 says that, assuming the strong arithmetic permanent vs. determinant and derandomization conjectures, one can construct an explicit family of linear separators of small dimension between $\Delta_V[g, m]$ and $f = \text{perm}^\phi(Y)$.

It has to be stressed that Theorem 9.2 critically depends on the exceptional nature of $f = \text{perm}^\phi(Y)$ and $g = \det(Y)$. If one were to consider general f and g in place of the permanent and determinant, the conclusion of Theorem 9.2 will almost never hold. For general f and g, a global obstruction set $S_{n,m}$ that gives a linear separator ψ between $\Delta_V[g, m]$ and f can be constructed (if it exists) by appropriately eliminating $\dim(V) - r$ variables. This can be done using general purpose algorithms in algebraic geometry for computing multivariate resultants and Gröbner bases. But these algorithms take $\Omega(\dim(V))$ space and $\Omega(2^{\dim(V)})$ time. Since $\dim(V)$ is exponential in n and m, the time taken is at least double exponential in n and m, and the total bit length of $S_{n,m}$ is exponential in n and m. Nothing better can be expected for general f and g, because elimination theory is in general intractable. Specifically, the problem of computing the Gröbner basis is EXPSPACE-complete [18]. This means it takes in general space that is exponential in the dimension of the ambient space, which is $P(V)$ here. In contrast, Theorem 4.2 says that a short specification $S_{n,m}$ of a linear separator between $\Delta_V[g, m]$ and $f = \text{perm}^\phi(Y)$, can be computed in $\text{poly}(n, m)$ time exploiting the exceptional nature of f and g. This may seem unbelievable.

At present, such explicit separators of small dimension can be constructed in algebraic geometry only between very special kinds of algebraic varieties, such as the Grassmanian or the flag varieties [9], and very special kinds of points. This can be done using the second fundamental theorem of invariant theory [9, 37] which gives a very nice explicit set of generators for the ideals of these varieties. But these varieties have very low complexity in comparison to $\Delta_V[n, m]$. For example, their complexity, according to a certain complexity measure on (quasi)-homogeneous spaces defined in [16], is zero, whereas that of $\Delta_V[g, m]$ is quadratic in m. Furthermore, they are normal, whereas $\Delta_V[g, m]$ is not normal according to a recent result [35]. The problem of explicit construction of linear separators when the underlying variety is not normal and its complexity is so high seems very formidable and far beyond the reach of the existing machinery in algebraic geometry. Theorem 9.2 says that such formidable explicit construction problems in algebraic geometry are hidden underneath the hardness and derandomization hypotheses in complexity theory.

10 Flip in the boolean setting

To get an efficient pseudorandom generator, it does not suffice to just assume that $P \neq NP$. One needs a stronger average case assumption, namely, existence of one way functions. Similarly, to get a flip theorem in the context of the usual (boolean) $NP \not\subseteq P/poly$ conjecture, one needs to assume a stronger average case form of this boolean conjecture based on characterization by symmetries. In this section we state this conjecture (Conjecture 10.3). The corresponding flip theorem (Theorem 10.5) then follows as a direct corollary of the main result in [12] on derandomization of BPP.

We begin with a preliminary motivating result in the context of the following strengthening of Conjecture 2.3.

Conjecture 10.1 *Analogues of Conjecture 2.3 and Conjecture 2.1 hold for any integral nonzero $e(X)$ with the properties E1' and E2 as in Theorem 5.1 (b).*

This gives a purely group-theoretic definition of hardness in the context of the arithmetic P vs. NP problem.

Theorem 10.2 (Flip for property E1) *Analogues of Theorems 4.2 and 4.3 hold for any nonzero integral $e(X)$ with the properties E1' and E2 (as in Theorem 5.1 (b)).*

This is proved just like Theorems 4.2 and 4.3, with Conjecture 10.1 playing the role of Conjecture 2.3 and the property E1' the role of E1.

Now we turn to the boolean setting. The following is a stronger form of the $NP \not\subseteq P/poly$ Conjecture.

Let S be the set integers of bit length at most n^3 (say). Let C be a boolean circuit whose input is the bit specification of X with entries in S. Let A be a co-RP algorithm for testing if C has properties E1' and E2 akin to the algorithm in the proof of Proposition 7.4 (for testing E1 and E2) with the following difference. Whenever we used a random number in that algorithm, we use a random integer of bitlength at most $n^3/3$, and instead of standard generators of $GL_n(\mathbb{C})$, we now use standard generators of $SL_n(\mathbb{C})$.

Conjecture 10.3 (Stronger invariant theoretic average case form of the $NP \not\subseteq P/poly$ conjecture) *Let C be any boolean circuit of $poly(n)$ bit size whose input is bit specification of X with entries in S. Suppose C comes with a* promise *that $prob\{C(X) = 0\}$, $X \in S$, is small, say $< 1/n$, where $C(X)$ denotes the boolean function computed by C.*

Then the algorithm A above for testing if C has properties E1' and E2 says NO with high probability ($\geq 1/poly(n)$).

The promise is necessary in Conjecture 10.3 since there exist small circuits with the properties E1' and E2 that are zero almost everywhere but not everywhere.

Proposition 10.4 *Conjecture 10.3 implies $NP \not\subseteq P/poly$.*

Proof: Let $E_b(X)$ be the boolean function which is zero if $E(X)$ is zero and one otherwise. Clearly $E_b(X)$ has properties E1' and E2. Furthermore, computation of $E_b(X)$ is NP-complete [10]. Hence it suffices to show that any boolean circuit computing $E_b(X)$ satisfies the promise. But the number of zeros of $E_b(X)$ is the same as those of $E(X)$. Hence by the Schwarz-Zippel lemma, $\text{prob}\{E_b(X) = 0\}$, $X \in S$, is bounded by $\deg(E(X))/|S| = n^{kn^2}/2^{n^3} < 1/n$. Q.E.D.

Conjecture 10.3 basically says that the symmetries E1' and E2 of $E_b(X)$ are hard to approximate on the average. This is an invariant theoretic average case form of the worst case assumption that $E_b(X)$ is hard to compute (as expected since it is NP-complete). It will be interesting to study the relationship (if any) between this average case assumption and the standard average case assumptions in complexity theory, such as existence of one way functions.

Theorem 10.5 (Flip in the boolean setting) *Suppose Conjecture 10.3 holds and also that the complexity class E does not have subexponential size circuits (or less stringently, that the co-RP algorithm A above can be derandomized in a black box fashion very much as in Section 7.4).*

Then for every n and $m = poly(n)$, it is possible to compute in $poly(n, m) = poly(n)$ time a small set $S_{n,m} = \{X_1, \ldots, X_r\}$, $r = poly(n, m) = poly(n)$, of $n \times n$ matrices with entries in S such that for every boolean circuit C satisfying the promise in Conjecture 10.3 and with total bit size $\leq m$ (and hence, in particular, for any boolean circuit of size $\leq m$ claiming to compute $E_b(X)$), $S_{n,m}$ contains a matrix X_C which is a counter example against C (as detected in the algorithm A).

Furthermore, assuming an appropriate stronger form (analogous to Conjecture 7.5) of the assumption that E does not have subexponential size circuits, (or less stringently, that the co-RP algorithm A above can be derandomized in a black box fashion) Conjecture 10.3, and hence, $NP \not\subseteq P/poly$ conjecture, has an explicit proof–i.e., there exists an obstruction family $\tilde{O}$ satisfying F0-F4–except that the obstructions are now only against small circuits satisfying the promise in Conjecture 10.3.

The new ingradient here is formulation of Conjecture 10.3, i.e., formulation of the conjecturally correct nonadaptive co-RP algorithm algorithm A for finding a counterexample against any small boolean circuit claiming to compute $E_b(X)$. Once this is done, Theorem 10.5 is just a direct corrollary of the main result in [12] on derandomization of BPP, because the algorithm A can be derandomized under standard assumptions therein. This algorithm A is to be contrasted with the *adaptive* probabilistic polynomial time algorithm in [3] for finding a counterexample against a small boolean circuit claiming to compute SAT, assuming $NP \not\subseteq P/poly$.

Let us finish this section with one more variant of a flip theorem.

Theorem 10.6 (Flip over a finite field) *Analogue of Lemma 4.1 holds over a large enough finite field F_p, $p > 2n$ (say), instead of $\mathbb{Q}$ or $\mathbb{C}$, provided in the definitions of F1-4 we confine ourselves to the circuits with the promise that the polynomials computed by them have the same degree as that of perm(X) (otherwise the circuit cannot compute perm(X) for trivial reasons).*

Similar analogue of Theorem 4.2 holds for hardness of the function $E(X)$ over a large enough F_p as in Conjecture 2.1.

This is also proved like Theorem 4.2.

11 Rigidity

The proof (cf. Section 7.3) of the flip Theorem 4.2 works for any function $e(X)$, $X = (x_1, \ldots, x_n)$, over $\mathbb{Q}$ in the place of $E(X)$ as long as $e(X)$ has the following properties:

(1) It is characterized by symmetries in the following sense:

Definition 11.1 *We say that $e(X)$ is characterized by symmetries if it is the only nonzero polynomial (up to a constant multiple) with rational coefficients that satisfies a small (poly(n)) number of algebraic polynomial identities with integral coefficients (in the spirit of those in the property (E)), each having a specification of poly(n) bitlength and containing $O(1)$ terms. Here each identity is of the form*

$$g(e(Y_1), \ldots, e(Y_k)) = 0,$$

where $g(u_1, \ldots, u_k)$ is a polynomial computable by a circuit over $\mathbb{Z}$ of $O(1)$ size with input u_i's, and each Y_i can be computed by a circuit over $\mathbb{Z}$ of poly(n) bit size with input X.

If we only require that each $g(u_1, \ldots, u_k)$ be computable by a poly(n) bit size circuit over $\mathbb{Z}$ with input u_i's, we say that $e(X)$ is weakly characterized by symmetries.

The circuits specifying the identities here can be nonuniform.

(2) $e(X)$ cannot be computed by an arithmetic circuit over $\mathbb{Q}$ of poly(n) bit size.

Here (1) implies that there is a nonadaptive co-RP/poly algorithm for deciding if a given arithmetic circuit C computes $e(X)$ (akin to that in the proof of Proposition 7.4), where a co-RP/poly algorithm means a nonuniform algorithm in the form of a poly(n) size circuit with random advice in addition to the usual input. Nonuniformity has to be allowed since the circuits specifying the identities in Definition 11.1 can be nonuniform. It is easy to see that the proof of the flip theorem goes through even in the presence of such nonuniformity. It also goes through even when $e(X)$ is required to be characterized by symmetries in a weaker sense, except that F1 (b) need not hold in this weaker setting.

Proposition 11.2 *The number of $e(X)$ over $\mathbb{Q}$ that are characterized by symmetries in a weaker sense (Definition 11.1) is $\leq 2^{poly(n)}$.*

Proof: This holds because the total bit length of the specification of the identities in Definition 11.1 in terms of small circuits is $O(\text{poly}(n))$. Q.E.D.

The proposition implies that the proof technique of the flip Theorem 4.2, which only works for functions with properties (1) and (2), is *extremely rigid*. By this we mean that it only works for $2^{\text{poly}(n)}$ number of functions in place of $e(X)$. This is also the case for Flip Theorem 10.5 in the boolean setting.

This form of rigidity is extremely severe in comparison to the *mild rigidity* constraint that the natural proof barrier [33] places [2] on proof techniques for the $NP \not\subseteq P/poly$ conjecture: namely, that they should work for less than $2^N/poly(N)$ number of functions, where $N = 2^n$ is the size of the truth-table specification of an n-ary boolean function.

It is a *plausible* that any proof of the arithmetic or boolean P vs. NP conjecture (or any of the related conjectures under consideration in this paper) has to be extremely rigid. This is because by Theorem 4.2 any proof of the (weak) arithmetic P vs. NP conjecture is close to an explicit proof. But the explicitness condition seems so severe that any proof that comes even close to an explicit proof may work for only rare exceptional functions (like the permanent or $E(X)$). That is, just mildly rigidity which suffices to bypass the the natural proof barrier [33]

[2]Ignoring the constructivity condition in [33]

may not be enough, and a proof may be forced to be extremely rigid, like that Theorem 4.2 or 10.5.

Acknowledgement:

The author is grateful to Janos Simon and Josh Grochow for helpful discussions.

References

[1] M. Agrawal, Proving lower bounds via pseudo-random generators, in proceedings of the FSTTCS, pages 92-105, 2005.

[2] S. Aaronson, Is P versus NP formally independent? Bulletin of the EATCS 81: 109-136 (2003).

[3] A. Atserias, Non-unifrom hardness for NP via black-box adversaries, ECCC, report 154 (2005).

5sequences, J. Combin. Theory Ser. A, 24 (1978), 296-307.

[4] A. Bogdanov, L. Trevisan, On worst case to average case reduction for NP problems, SIAM J. Comput., vol. 36, No. 4, pp. 1119-1159.

[5] D. Eisenbud, Commutative algebra with a view toward algebraic geometry, Springer-Verlag, 1995.

[6] J. Feigenbaum, L. Fortnow, Random self reducibility of complete sets, SIAM J. Comput., 22 (1993), pp. 994-1005.

[7] L. Fortnow, A. Pavan, S. Sengupta, Proving SAT does not have small circuits with an application to the two queries problem, JCSS 74 (2008), 358-363.

[8] G. Frobenius, Uber die Darstellung der endlichen Gruppen durch lineare Substitutionen, Sitzungsber Deutsch. Akad. Wiss. Berlin (1897), 994-1015.

[9] W. Fulton, J. Harris, Representation theory, A first course, Springer, 1991.

[10] L. Gurvits, On the complexity of mixed determinants and related problems, pp. 447-458, Lecture notes in computer science, Springer Verlag, September, 2005.

[11] O. Ibarra, S. Moran, Probabilistic algorithms for deciding equivalence of straight line programs, JACM 30 (1), 217-228.

[12] R. Impagliazzo, A. Wigderson, $P = BPP$ unless E has sub-exponential circuits: Derandomizing the XOR lemma, Proceedings of the 29th STOC, 1997.

[13] V. Kabanets, R. Impagliazzo, Derandomizing polynomial identity tests means proving circuit lower bounds, Computational Complexity, 13(1-2), pages 1-46, 2004.

[14] J. Landsberg, L. Manivel, N. Ressayre, Hypersurfaces with degenerate duals and the Geometric Complexity Theory Program, arXiv:1004.4802, April, 2010.

[15] S. Lang, Algebra, Addison-Wesley, 1997.

[16] D. Luna, Th. Vust: Plongements d'espaces homogenes. Comment. Math. Helv. 58, 186 (1983).

[17] M. Marcus, F. May, The permanent function, Canad. J. math., 14 (1962), 177-189.

[18] E. Mayr, and A. Meyer, The complexity of the word problems for commutative semigroups and polynomial ideals, Advances in mathematics, 46 (3): 305-329, 1982.

[19] T. Mignon, N. Ressayre, A quadratic bound for the determinant and permanent problem, International Mathematics Research Notices (2004) 2004: 4241-4253.

[20] K. Mulmuley, M. Sohoni, Geometric complexity theory, P vs. NP and explicit obstructions, in "Advances in Algebra and Geometry", Edited by C. Musili, the proceedings of the International Conference on Algebra and Geometry, Hyderabad, 2001.

[21] K. Mulmuley, On P vs. NP, and geometric complexity theory, technical report, computer science department, the university of Chicago, July 2010. Available at: http://ramakrishnadas.cs.uchicago.edu.

[22] K. Mulmuley, M. Sohoni, Geometric complexity theory I: an approach to the P vs. NP and related problems, SIAM J. Comput., vol 31, no 2, pp 496-526, 2001.

[23] K. Mulmuley, M. Sohoni, Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties, SIAM J. Comput., Vol. 38, Issue 3, June 2008.

[24] K. Mulmuley, M. Sohoni, Geometric complexity theory III, on deciding positivity of Littlewood-Richardson coefficients, cs. ArXiv preprint cs. CC/0501076 v1 26 Jan 2005.

[25] K. Mulmuley, M. Sohoni, Geometric complexity theory IV: quantum group for the Kronecker problem, cs. ArXiv preprint cs. CC/0703110, March, 2007.

[26] K. Mulmuley, H. Narayanan, Geometric complexity theory V: on deciding nonvanishing of a generalized Littlewood-Richardson coefficient, Technical Report TR-2007-05, computer science department, The University of Chicago, May, 2007. Available at: http://ramakrishnadas.cs.uchicago.edu

[27] K. Mulmuley, Geometric complexity theory VI: the flip via positivity, revised version, technical report, computer science department, The University of Chicago, July 2010. Available at: http://ramakrishnadas.cs.uchicago.edu

[28] K. Mulmuley, Geometric complexity theory VII: Nonstandard quantum group for the plethysm problem, Technical Report TR-2007-14, computer science department, The University of Chicago, September, 2007. Available at: http://ramakrishnadas.cs.uchicago.edu.

[29] K. Mulmuley, Geometric complexity theory VIII: On canonical bases for the nonstandard quantum groups, Technical Report TR 2007-15, computer science department, The university of Chicago, September 2007. Available at: http://ramakrishnadas.cs.uchicago.edu.

[30] D. Mumford, Algebraic geometry I: complex projective varieties, Springer, 1976.

[31] D. Mumford, J. Fogarty, F. Kirwan: Geometric invariant theory. Springer-Verlag, 1994.

[32] N. Nisan, A. Wigderson, Hardness vs. randomness, J. Comput. Sys. Sci., 49 (2): 149-167, 1994.

[33] A. Razborov, S. Rudich, Natural proofs, J. Comput. System Sci., 55 (1997), pp. 24-35.

[34] J. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, Journal of the ACM 27, 1980.

[35] Shrawan Kumar, Geometry of orbits of permanents and determinants, arXiv:1007.1695v1.

[36] L. Valiant, The complexity of computing the permanent, Theoretical Computer Science 8, pp 189-201, 1979.

[37] H. Weyl, Classical groups. Their invariants and representations, Princeton University Press, 1939.